

FIG. 14 is a linear feedback register 290 used to calculate an icon (CRC, polynomial code) in accordance with one embodiment of the invention. The icon generator 290 has a data register (shift register) 292 and an intermediate remainder register 294. The specific generator of FIG. 14 is designed to calculate a cyclical redundancy code (CRC-16). The plurality of registers 296 in the intermediate remainder register 294 are strategically coupled by a plurality of exclusive OR's 298. The data bits are shifted out of the data register 292 and into the intermediate register 294. When the data bits have been completely shifted into the intermediate register 294, the intermediate register contains the CRC associated with the data bits. Transform generators have also been encoded in software.

FIG. 15 is a block diagram of a system 300 for associative processing in accordance with one embodiment. The system 300 has multiple IG/APUs (icon generator/associative processing units; plurality of icon generators; plurality of associative processing units) 302, 304, 306. The IG/APUs 302, 304, 306 have an input connected to key data or input data streams 308, 310, 312. The IG/APUs are connected to a bus (network or inter-processor communication bus) 314. An AMC 316 is also connected to the bus 314. Generally, only icons of fixed length are passed over the bus 314. This significantly reduces the bus traffic and therefor the required bandwidth of the bus. The AMC 316 is connected to RAM 318 containing a database in one embodiment.

Thus there has been describe a system for associative processing that may be configured to perform any number of tasks including,

associative databases, content scanning, packet accounting, extensible markup language database management systems and more.

FIG. 16 is a block diagram of a system 330 for implementing behavioral operations in accordance with one embodiment of the invention. The system 330 has a search engine 332. The search engine 332 is connected to an associative match memory 334. A behavioral operation unit 336 is connected to the associative match memory 334. The operation of a search engine is explained with respect to figures 1-6. The search engine can be implemented in software (firmware) or may be implemented in hardware. The behavioral operation unit 336 is implemented in memory and defines the behavior of the search engine 332.

FIG. 17 is a block diagram of a system 340 for implementing behavioral operations in accordance with one embodiment of the invention. An icon generator 342 is connected to a key data fetch unit 344. The key data fetch unit 344 is connected to the input data 346. The icon generator 342 is connected to an associative processing unit (APU) 348. The APU 348 is connected to the associative memory controller (AMC) 350. The AMC 350 is connected to RAM 352 which stores quanta 354. The quanta 354 may contain an association 356, a behavioral flag (behavioral indicator) 358, field description numbers 360 and other information. The RAM 352 is connected to the field descriptor array 362. The RAM 352 is connected to an association stack 364. The AMC 350 is connected to an execution stack 366. The APU 348 is connected to the behavioral operation unit 368.

When the AMC 350 locates an association 356, one or more behavioral flags 358 are encountered. The AMC 350 receives the behavior flags 358 and the field descriptor 360 for processing. The APU 348 causes a new key data that is specified by the field data to be
5 fetched by the key data fetch unit 344. The key data is then iconized by the IG 342. The APU 348 then executes the specific behavior specified by the behavioral flags 358. When a quanta contains a particular behavior flag it is said to belong to the set of quantas that have that behavior or belongs to a behavioral set. Behaviors are generally
10 accommodated (implemented) by the use of logical operators, state machines or both. When a behavior flag is set, the corresponding behavior operational unit is activated. Certain behavior combinations are supported, so multiple behavioral operation units can be activated at the same time. Some behaviors involve iconizing new key data using the
15 quantas's field descriptor to locate the key data. A field descriptor consists of a list of byte offsets and a mask for "don't care" bits.

There are two stacks in the system 340. The association stack 364 is used to hold possible association return values. For some operations, there is no way to determine which association to return until an
20 association thread has been completed. For example, it is possible to have quanta that indicates it contains the return association (so it is pushed on the stack) unless a "better match" is found. The quanta would also contain a behavior flag that tells the APU how to go about finding a better match. If a better match is subsequently found, its return
25 association value is pushed on the stack. Another behavior, for example, indicates than an exception to the current match condition may exist. If